

Optics vs. Electronics in Future High-Capacity Switches/Routers

*Original*

Optics vs. Electronics in Future High-Capacity Switches/Routers / Bianco, Andrea; D., Camerino; D., Cuda; Neri, Fabio. - STAMPA. - (2009), pp. 1-6. (Intervento presentato al convegno IEEE Workshop on High Performance Switching and Routing (HPSR 2009) tenutosi a Paris, France nel June 22-24, 2009) [10.1109/HPSR.2009.5307441].

*Availability:*

This version is available at: 11583/2285599 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/HPSR.2009.5307441

*Terms of use:*

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# Optics vs. Electronics in Future High-Capacity Switches/Routers

Andrea Bianco, Daniela Camerino, Davide Cuda, and Fabio Neri.

Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy

Email: {andrea.bianco, davide.cuda, daniela.camerino, fabio.neri}@polito.it

**Abstract**—The rapid growth of the Internet traffic demand is bringing the current electronic switches closer to their intrinsic technological limits, which may be overcome by optical technologies. Although optical packet switching has been widely studied in academia; the limited processing and buffering capabilities available in the optical domain makes the implementation of an all optical packet switch difficult in practice. We considered a hybrid (electro/optic) switching architecture, relying on an all-optical switching fabric, aiming to understanding the trade-offs between current photonic and electronic technologies. We consider both the classical fixed-size synchronous paradigm and the asynchronous operation mode, which may suit better to the optical technologies. Buffers organized according to either an electronic FIFO (First Input First Output) paradigm or a photonic FDL (Fiber Delay Line) mechanism are analyzed and compared.

**Keywords**—all-optical switch

## I. INTRODUCTION

Due to the increasing number of new services and applications like Voice over IP (VoIP), high-definition video broadcasting, video conferencing and peer to peer applications, Internet traffic keeps growing. Conservative measurements show that bandwidth demand roughly grows at a pace of about 50% every 18 months [1], [2], hence at the order, or even faster than Moore's law.

Although at the transmission layer, the optical technologies, in particular Wavelength Division Multiplexing (WDM) techniques, emerged as the winning approach, switching nodes are still mainly based on electronic technologies. Nowadays, high-end switches/routers must still perform Optical-to-Electronic-to-Optical conversion to process all the traffic for switching/routing. This implies that continuously increasing data rates, are bringing switches/routers closer to intrinsic limits of the electronic technology; indeed, each new routers' generation requires more complex control algorithms, consumes more energy and dissipates more power than the previous one.

Hence, in order to bridge the gap between the capacity offered by the optical transmission systems and the electronic processing capability, both industry and academia agree that some switching functions should be simplified or moved from the electrical to the optical domain. Efforts to shift some functionalities from the electronic to the optical domain have already been done and some prototypes, employing the optical technology, have been implemented for processor interconnection, as in the case of the recent PERCS (Productive, Easy-to-use, Reliable Computing System) [4] and OSMOSIS (Optical

Shared Memory Supercomputer Interconnect System) [5]. The use of photonic technologies for switching is interesting; huge available bandwidth, reduced power consumption and dissipation, much larger information densities, a wavelength switching cost quite independent of the data bit-rate (differently from the electronic domain), are some of the advantages of the optical technology. Despite all these advantages, implementing a fully optical packet switch is far from being convenient today. Indeed, the lack of optical memories and the limited processing capabilities in the optical domain make it very difficult to solve conflicts in time domain through dynamic operations, which underlies the packet switching concept. Thus, the faith in all-optical packet switching has probably vanished, and several researchers are studying today hybrid electro-optical packet switching architectures.

In this context, the main challenge is in finding the best balance of optical and electronic technologies within a packet switching device that, externally, provides legacy (Ethernet, SDH) interfaces, over which IP packets are received and transmitted. The lack of commercially feasible 3R optical regenerators impairs a true digital operation in the optical domain. Fast optical switching can be obtained with Semiconductor Optical Amplifiers (SOAs) used as on-off gates, or even with more advanced components such as ring resonators [3]. However, other functionalities, necessary in IP routers or Ethernet switches, are difficult to implement in the optical domain. For example, segmentation of variable-size packets into fixed-size data units must be done electronically. Even alignment of fixed-size packet for a synchronous switch operation is hard in optics. Finally, a good equivalent of Random Access Memories (RAMs) is completely missing. Since packet alignment/synchronization operations are difficult to achieve in the optical domain, asynchronous switching of variable-size packets should be given serious attention; therefore, both the synchronous switching operation, working with fixed-size packets, and the asynchronous variable size packet paradigm are considered in this paper.

Contention resolution in packet switching is done in a distributed fashion exploiting temporary storage of contending packets in RAMs. Input/output bandwidth of electronic memories does not scale easily to high data rates, and relatively complex packet storage structures are required, like Virtual Output Queues (VOQs) in Input Queued (IQ) switches. Indeed, VOQs include elementary switching functionalities themselves, as incoming packets must be routed to the proper

storage space, thus creating scalability problems for very high speed links and large switch port counts. Optical switching fabrics could provide large bit-rate speed-ups with respect to external line rates. However, exploiting this speed increase is difficult in practice, mainly due to bandwidth limitations of input/output packet buffers. Therefore, novel switch designs are needed, in which the features of optical switching fabrics can be exploited without demanding extreme performance to packet buffers. Up to now, Fiber Delay Lines (FDL) are the only approach available to buffer information in the photonic domain [6]. However, switched FDLs have large physical size, are difficult to control and costly to implement. Conversely, electronic buffers can become very costly and power hungry at very large data rates; hence, they should be parsimoniously used, and shared, if possible, within the switching architecture. Furthermore, complex queueing architectures should be avoided. Thus, we wish to contrast FDL-based solutions with simple FIFO (First Input First Output) electronic buffering schemes, to understand the merits of the two approaches.

The paper is organized as follows. In Sec. II we describe the architectures under study, considering either FDL or simple FIFO electronic buffers, both considering the synchronous and the asynchronous operation mode. In Sec. III, performance of the different architectures is compared by simulation. Finally, we draw some conclusions and guidelines for future work in Sec. IV.

## II. THE SWITCHING ARCHITECTURE

Fig. 1 shows the considered architecture. This architecture is difficult to characterize with respect to the taxonomy used to classify classical electronic architectures presented in the literature. It represents an intermediate solution between an Input Queue (IQ) and an Output Queue (OQ) switch. OQ switches achieve 100% throughput (see [7]), but require a speed-up in both the switching fabric and the memory access speed. The internal speed increase with respect to the I/O line rate is equal to  $N$ ,  $N$  being the number of inputs/outputs. Thus, due to technological constraints, OQ architecture are not easily scalable to high-speed, large-size, switches.

On the contrary, IQ switches do not require any internal speed increase, since both the switching fabric and the memories run at line speed. However, they either show limited performance or require rather complex queue architectures and scheduling algorithms to enhance performance. Both synchronous and asynchronous IQ switches suffer from the Head of the Line (HoL) blocking problem. Indeed, the maximum achievable throughput for a IQ switch with a single FIFO at each input is equal 58% and to 51% under i.i.d. uniform Bernoulli traffic, for the synchronous [7] and asynchronous [8] case, respectively. The HoL issue can be solved in IQ synchronous switches employing Virtual Output Queue (VOQ) and using the Maximum Weight Matching (MWM) scheduling algorithm [9]. However, this solution requires a non scalable queue architecture: the queue control complexity increases linearly with the line speed, since a routing decision must be performed to store data into the proper input queue. Thus,

we will avoid using VOQs in our architecture. In order to preserve scalability, we need to look for feasible solutions when considering very high bit rate on I/O lines. The switch architecture depicted in Fig. 1, which has been previously proposed for all-optical switches [10], comprises  $N$  input and output ports, which, for simplicity, are assumed to operate at the same speed.  $M$  re-circulating lines including buffers are available. Thus, the switching fabric has  $N + M$  input/output lines.

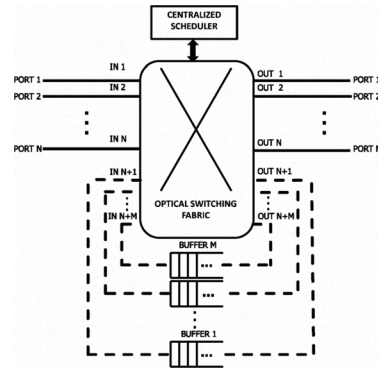


Fig. 1. Considered switching architecture.

The packet switch is built around an all-optical non-blocking switching fabric, which may be implemented using SOA gates or other photonic technologies. This helps in reducing the required energy and the power dissipation; furthermore, the switching complexity does not depend on the transmission bit rate. Assuming that the optical fabric scales well in the number of I/O lines (also thanks to the possibility of larger distances and physical sizes in the optical domain), we use  $M$  extra lines to interconnect buffers, thereby providing a form of spatial speed-up with respect to the  $N \times N$  fabric. The overall switch capacity is increased from  $N$  to  $N + M$  packets per packet time; hence, a  $(N + M)/N$  speed-up is offered.

$M$  ranges from 0 (buffer-less switch) to  $M = 2N$ , providing a spatial speed-up ranging from 1 to 3. Indeed, when  $M = 2N$  three output lines are, on average, available for each input port. At the same time, three input lines exist for each output port. However, unlike in traditional electronic switches, this speed-up is neither required in input/output lines data rate nor in memory access speed. The spatial speed-up allows the centralized scheduler to select among  $M + N$  (instead of  $N$ ) available packets at input lines, increasing the set of possible choices. Furthermore,  $M + N$  lines (instead of  $N$ ) are also available to switch packets arriving at the  $N$  input ports, temporarily reducing losses due to contention. Finally, to keep the architecture simple and to control the (electronic) processing overhead, we wish to use a simple queueing architectures. Thus, the re-circulation lines lead to either electronic FIFOs or photonic FDLs.

### A. Switch configurations

Although the basic switch architecture was defined, several design choices are still available, leading to a set of possi-

ble switch configurations. We discuss the possible choices for: i) the switching operation modes (synchronous vs. asynchronous), ii) the storage technologies used in the re-circulation buffers (photonic FDL vs. electronic FIFO) and iii) the buffer management policies (dedicated vs. shared).

In all architectures, we assume that a centralized scheduler selects packets to be transferred to output ports. When contention arise, i.e., more than one packet is willing to reach the same output port, priority is always given by the scheduler to packets available in the re-circulating buffers. One among all contending packets at the head of the re-circulating buffers is chosen according to a round robin scheme. Giving priority to buffered packets reduces or brings to zero the probability of out-of-order delivery.

We first describe the synchronous and asynchronous switching modes.

- In the case of *synchronous* (SYN) operation, the switch deals with time-aligned fixed-size data (named cells). Considering that most of the traffic is constituted by variable size IP datagrams, when packets are segmented into fixed-size cells, some bandwidth is wasted mainly due to two effects: (i) un-filled cells and (ii) additional control information required to re-assemble packets. Indeed, the last cell generated by a packet segmentation process is in general only partially filled, if the packet size is not an integer multiple of the cell size. Furthermore, the packet overhead is increased since each cell must transport control information (such as cell sequence number, last-cell flag, packet identifier, payload size, and possibly routing information) to permit a proper packet routing in the switching fabric and reassembly at switch outputs. Finally, one of the main drawbacks of SYN architectures is the need to distribute a precise clock signal among all cards, and to time-align cell at switch inputs (which requires buffering, and is costly in the optical domain) to ensure synchronous behavior.

In SYN operation, the scheduler makes scheduling decisions at cell boundaries. When cells arrive at input ports, if there is no contention, i.e., all cells are willing to reach a different output port (considering cells both at input ports and at the head of re-circulation buffers), they are synchronously forwarded to the proper output. In case of output contention, one cell is forwarded to the proper output, and all other contending cells are switched to free available re-circulation buffers, if any. Otherwise, cells are dropped.

- In the case of *asynchronous* (ASY) operation mode, the scheduler does not operate on a time slotted basis. Although asynchronous switches have been traditionally considered as architectures that provide worse throughput performance, they offer two significant advantages, especially in the context of optical switching. First, no time reference signal distribution among line-cards and the switching fabric, nor time-alignment of input cells, are required. Indeed, these operations are increasingly complex for increasing transmission speed (remember

that packet durations shrinks to the nanosecond scale as line bit-rates approach the Tb/s figure) and switch sizes, and requires a significant amount of power consumption. Second, variable-size packet switching with in-order delivery can be easily supported with no segmentation at inputs and, especially, without re-assembly procedures at outputs; hence, there is no need for Output Reassembly Machines (ORMs). Moreover, the control information required in this case is smaller than the one required by a SYN switch, since only some routing information may have to be added to packets.

In the case of ASY switching, the centralized scheduler does not work on a time slotted basis. The events that trigger the scheduler are the end of a packet transmission on an output port or the arrival of a packet at input lines when output lines are idle. When an output becomes available, if there is at least one packet addressed to this output (either at input ports or at the re-circulation buffer), a new packet is transmitted. When a packet arrives at a given input port either it is immediately transmitted to the proper output port, if available, or it is stored into a buffer, if available; otherwise it is discarded.

Both all-optical packet switching paradigm as well as an hybrid electro-optical solution can be supported. In the former case, buffers are implemented by means of FDL, while in the latter case, a hybrid architecture is obtained by using high-speed electronic FIFO buffers. The choice of the FIFO scheduling discipline being driven by the need of keeping low the electronic complexity.

- *Fiber Delay Line* (FDL). FDLs are based on a fiber that delays the optical packet by a fixed amount of time related to the propagation delay within the fiber. From the scheduler point of view, FDLs bring back to the input lines a stored packet after a given fixed delay  $\Delta$ . Many studies assume that optical packets can re-circulate many times in FDLs. However, this solution requires packet regeneration, increases the architecture complexity, and introduces an additional delay. Therefore, this possibility is not considered further in this paper. Thus, when a packet is re-directed to a FDL, it becomes again available at input lines after a delay equal to  $\Delta$ . Either it is transmitted immediately, if the proper output is free, or it is discarded. When FDLs are used, the switching fabric is subject to the void filling problem. Indeed, even if an output port is free and there is a packet addressed to it traveling across the FDL, it can not be transmitted until it is brought back to the input line.
- *Electronic FIFO*. In this case, buffers are implemented via RAMs with a FIFO service discipline. We use a simple FIFO and not a more complex structure, as VOQ, because we want to keep the proposed architecture as simple and as scalable as possible. Indeed, the optical switching fabric has the interesting feature of being quite independent of the bit-rate. In the electronic domain, FIFOs scale very well (linearly) with the line bit-rate. Moreover, FIFOs

do not require any complex management. The advantage with respect to FDLs lies in the fact that packets can be extracted by the scheduler without any timing constraint.

Regardless of the re-circulation buffers technology (FDL or RAM), a strategy to share the available packet buffers has to be defined. In this paper we analyze two possible solutions: buffers can either be shared by all ports or part of the buffers can be assigned in a pre-defined way to an output port.

- *Dedicated buffers* policy. Each buffer is associated with an output line. We consider the case  $M = N$ , i.e., the number of re-circulating buffers is equal to the number of input/output ports. An arriving packet that finds the desired output busy is switched to the buffer headed for that output, if the buffer is not full; otherwise, the packet is dropped. Therefore, the switch can solve contentions only among two packets (one packet at an input line, one packet in one buffer).
- *Shared buffers* strategy. In this case, buffers are shared by all inputs, regardless of packets destination. Therefore, out-of-sequence delivery may occur, and re-ordering of packets must be provided at the output ports to avoid this phenomenon. A packet arriving at an input port that finds the proper output busy is routed to the buffer with minimum occupancy among those to which no other packet is currently being transferred (i.e., the output line leading to the buffer is free). Buffer sharing is expected to be more effective in contention resolution. Indeed, if  $M \geq N$ , no contention can arise among packets at input ports, if there is enough space to store packets.

In both the considered buffer-management policies, note that no speedup is provided at the fabric output lines leading to packet buffers. Hence, only one packet at a time can be routed to a given buffer. When more than one input packet should be stored in a given buffer at the same time, contention arises and some contending packets must be discarded. This obviously introduces a form of blocking in the switching architecture.

### III. PERFORMANCE EVALUATION

We present performance results obtained by simulation considering the architecture described in Sec. II. The number of inputs/outputs ports  $N$  is set to 16, each port is running at 10 Gbps. The number of re-circulating lines  $M$  is set equal either to 16, in the case of dedicated buffers, or to 8, 16 and 32 in the case of shared buffers.

We assume a uniform traffic pattern, i.e., each output has the same probability  $1/N$  to be selected each time a packet is generated. To model a bursty packet generation process we use the classical ON-OFF process. The average packet length  $E[L]$  is equal to 576 Bytes, the average packet length emerging today from Internet traffic measurements [11]. We denote by  $B_r$  the transmission rate of input/output links. To obtain the desired average load  $\rho$ , ON and OFF times are chosen as follows:

$$E[T_{\text{ON}}] = \frac{E[L]}{B_r} \quad (1)$$

$$E[T_{\text{OFF}}] = E[T_{\text{ON}}] \times \frac{1 - \rho}{\rho} \quad (2)$$

- ASY traffic: both ON and OFF periods are exponentially distributed, with mean  $E[T_{\text{ON}}]$  and  $E[T_{\text{OFF}}]$ .
- SYN traffic: both the ON period (packet length in cells) and the OFF period follow a geometric distribution. The cell size is set to 64 Bytes; thus, an average packet length is equal to 9 cells. If collisions arise and a cell cannot be stored, the whole packet is lost.

Delays are normalized to the average packet size: a delay equal to 1 corresponds to  $0.460 \mu\text{s}$  (the time required to transmit 576 Bytes at 10 Gbps).

FIFOs have a storing capacity equal to  $K$  packets of average size. To fairly compare FDLs and FIFOs, the delay  $\Delta$  is set to be equal to  $K$  times the time needed to transmit an average packet size at the port line rate; implying that a FDL line can store up to  $K$  average-size packets sequentially.  $K$  is equal to 20 in our simulations. Both SYN (fixed-size packets time-aligned at inputs) and ASY (variable-size packets arriving at independent time) switch operations are considered.

We also consider as a reference a buffer-less switch ( $M = 0$ ). Indeed, under Bernoulli traffic the buffer-less switch performance can be easily computed [7].

Simulation runs exploits a proprietary simulation environment developed in C language. Statistical significance of the results are assessed by running experiments with an accuracy of 2% under a confidence interval of 95%.

#### A. Dedicated buffers

Fig. 2 shows the average delay vs. throughput when dedicated buffers are adopted: both FDLs and FIFOs are considered. In Fig. 2, SYN (ASY) architectures are identified by white (black) markers, whereas dashed (solid) lines identify FDL (electronic FIFO).

First, electronic FIFOs ensure higher throughput, but also larger delays. Indeed, when FDLs are used, both in the synchronous and asynchronous case, packets cannot be delayed for more than  $\Delta$  seconds. Afterwards, buffered packets/cells are either successfully transmitted (if no contentions occurs) or discarded, since only one FDL circulation is allowed. Conversely, when FIFOs are employed, packets/cells can be stored until their destination becomes available.

Electronic FIFOs increase throughput performance by roughly 20% if compared to FDLs. When FIFOs are employed, losses are only due to packets contentions. When more than two input packets have the same output destination, only one packet can be transmitted, one can be stored into the FIFO and all other packets are lost. Besides contention, FDL presents two additional problems. First, packet contentions can persist. Indeed, an arriving packet suffering contention is directed to the proper FDL; hence, the same packet is brought back at input lines after a fixed delay  $\Delta$ . Thus, if the packet does not find the output port available due to contention, it must be dropped. Second, bandwidth is wasted because of the void filling problem. Suppose an output port becomes available after a successful packet transmission. Even if there is a packet

traveling in the FDL willing to reach this output, the scheduler must wait before being able to attempt a new transmission, leaving the switch port idle and wasting bandwidth.

Fig. 3 shows the average and maximum FIFO occupancy (values are normalized to the buffer size). On average, queues are relatively empty and the maximum occupancy is relatively high, even though it never exceeds the buffer size; thus, performance are mainly limited by contentions and not by buffer overflow.

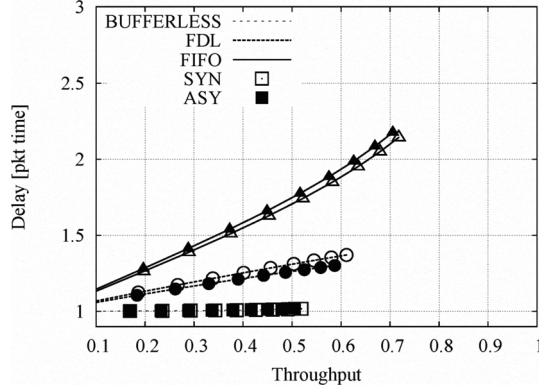


Fig. 2. Delay vs. throughput when buffers are managed using the dedicated strategy;  $N = M = 16$ .

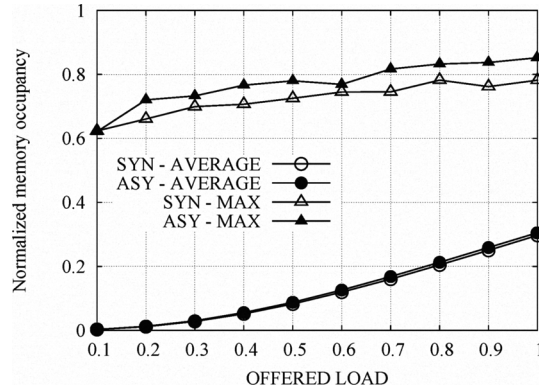


Fig. 3. Memory occupancy when buffers are managed using the dedicated strategy;  $N = M = 16$ .

### B. Shared buffers

Fig. 4 shows delay vs. throughput performance when the shared buffer strategy is adopted and FIFOs are employed. White (black) marks refer to the SYN (ASY) operation mode, respectively. Compared to the dedicated strategy, the shared policy ensures higher throughput for both SYN and ASY switching modes. Indeed, with  $M = N = 16$ , throughput increases from 0.7 to 0.8, and throughput close to 0.9 can be achieved with only  $M = 2N$  buffers (i.e. a spatial speed-up of 3). When the shared buffer policy is employed, packets are discarded only when buffers are completely full. Indeed, if more packets aim at the same output, one packet is transmitted, while the other packets select the buffer with minimum occupancy among available buffers (buffers to which no packet is

already being transferred). Delay increases because the number of contentions increases; thus, packets spend more time in memory. A contention may occur not only among inputs and the buffer associated with a given output, as in the dedicated buffer strategy, but between any input and any buffer. Finally, the SYN architecture shows better performance with respect to the ASY one.

Fig. 5 shows the results when FDLs are employed. Also in this picture, white (black) markers refers to the SYN (ASY) operation mode, respectively. The shared buffer strategy shows no significant performance gain; Thus, when FDL are employed, contention persistence and the void filling problem limit the switch throughput, independently of the policy used to manage buffers. Since packets either are transmitted after  $\Delta$  seconds or are discarded; delays are low and the performance gap between SYN and ASY switching modes reduces.

Fig. 6 and Fig. 7 show the average and the maximum queue occupancy respectively. Values are normalized to the buffer dimension. When the number of FIFOs is smaller than the number of input ports ( $M = 8$ ), SYN and ASY switching modes exhibits different behavior. In the ASY case, performance is limited by the buffer space (buffers are already completely filled up when the input load is around 0.5), while in the SYN case, contentions are the main limiting factor. Indeed, even if the maximum occupancy is reached, as shown in Fig. 7, Fig. 6 shows that the average queue occupancy is quite low. When the number of buffers is equal to, or larger than, the number of inputs, input contention disappears and performance are limited by the buffer size. As shown in Fig. 7, queues saturate first in the ASY case. Finally, the shared policy ensures a better exploitation of the memory space with respect to the dedicated buffer policy.

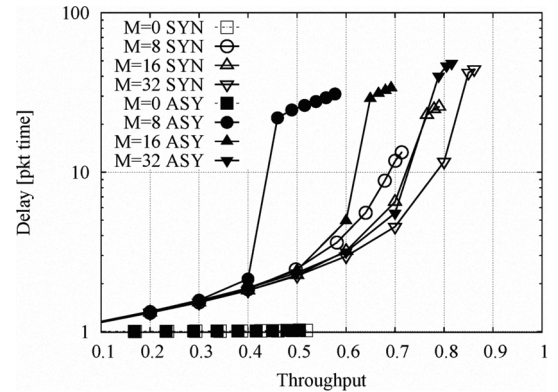


Fig. 4. Delay vs. throughput with FIFO buffers managed according to the shared strategy.

## IV. CONCLUSIONS

We studied a switch architecture exploiting electronic FIFO re-circulation buffers and we compared its performance with the classical “all-optical” architecture, where buffers are implemented by mean of FDLs. For both FDLs (optical) and FIFO (electrical) packet buffers, we considered synchronous and asynchronous switch operation modes. Moreover, two

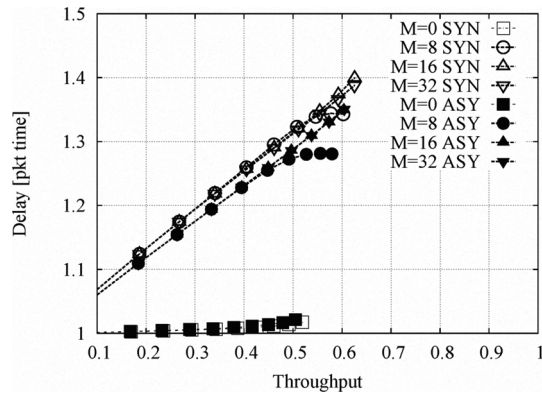


Fig. 5. Delay vs. throughput delay with FDL managed according to the shared strategy.

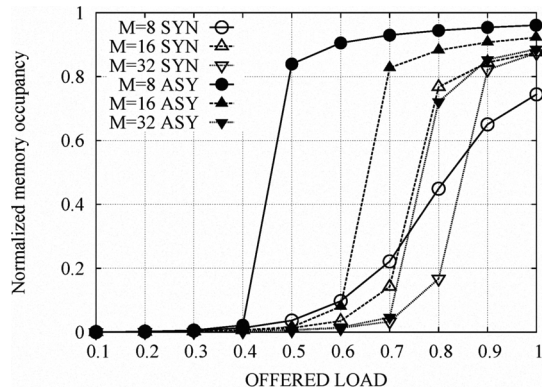


Fig. 6. Normalized average memory occupancy for the shared buffer strategy.

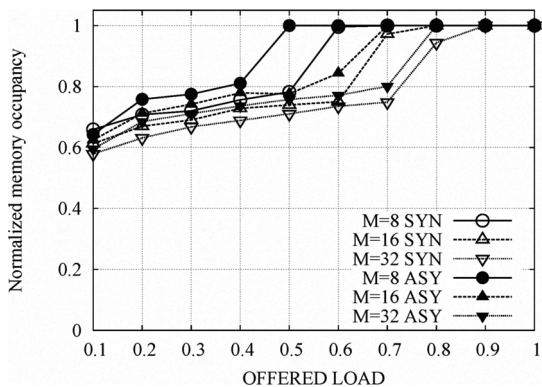


Fig. 7. Normalized maximum memory occupancy for the shared buffer strategy.

different strategies to manage buffers (dedicated vs. shared buffers) were proposed and analyzed.

The considered switch architectures, built around an optical switching fabric, exhibits interesting performance while keeping low control and implementation complexity. Interesting complexity/scalability/performance trade-offs emerge from our analysis, together with indications on which switch functionalities can be simplified or moved from the electrical to the optical domain.

In the considered architecture, contention resolution de-

pends less heavily than in current switches from the availability of large and complex packet buffering structures. Contention resolution is shifted from the time domain (i.e., from massive packet storage) to the space domain (i.e., extra ports and interconnections in the switch), a more suitable solution for photonic realizations.

Finally, Electronic FIFOs always exhibits better performance (especially if the shared strategy is adopted), but have complexity and power requirements that increase with the line rate, and are less scalable. The ASY switching mode is simpler to implement and avoids several costly functions (time-alignment of input packets, segmentation and reassembly). While it shows worse performance than the SYN operation, our results also show that it does not penalize significantly switch performance with respect to the SYN case. Thus, it may be considered as a more promising architecture, given its simplicity and the ubiquity of variable-size IP datagrams; even considering that minor throughput losses should not be considered as a major issue in the context of optical technologies where huge amount of bandwidth available.

#### ACKNOWLEDGMENTS

This work was partially supported by the BONE research project, a Network of Excellence funded by the European Commission within the 7th Framework Programme.

#### REFERENCES

- [1] Minnesota Internet Traffic Studies, available at <http://www.dtc.umn.edu/mints/home.php>
- [2] AMS-IX Traffic Statistics, available at <http://www.ams-ix.net/technical/stats/>
- [3] A.Shacham, K.Bergman, L.Carloni, "Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors", IEEE Transactions on Computers, vol.57, n.9, Sep.2008.
- [4] K.J. Barker, et al. "On the Feasibility of Optical Circuit Switching for High Performance Computing Systems", ACM/IEEE Conference on Supercomputing (SC '05), Seattle, WA, Nov. 2005.
- [5] R. Hemenway, R. Grzybowski, C. Minkenberg, and R. Luijten, "Optical-packet-switched interconnect for supercomputer applications", Journal of Optical Networking, Vol. 3, Issue 12, pp. 900-913.
- [6] C. Develder, M. Pickavet, and P. Demeester, "Choosing an appropriate buffer strategy for an optical packet switch with a feed-back FDL buffer", ECOC 2002, Sept. 2002, Copenhagen, Denmark.
- [7] M.Karol, M.Hluchyj, S.Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch", IEEE Trans. on Communications, vol.35, n.12, pp.1347-1356, Dec.1987.
- [8] Y.Oie, M.Murata, K.Kubota, H.Miyahara, "Performance Analysis of Nonblocking Packet Switch with Input and Output Buffers", IEEE Transactions on Communications, vol.40, n.8, pp.1294-1297, Aug. 1992.
- [9] N.McKeown, A.Mekkittikul, V.Anantharam, J.Walrand, "Achieving 100% Throughput in an Input-Queued Switch", IEEE Transactions on Communications, vol.47, n.8, pp.1260-1267, Aug. 1999.
- [10] J.Diao, P.L.Chu, "Analysis of Partially Shared Buffering for WDM Optical Packet Switching", JLT, vol.17, n.12, pp.2461-2469, Dec. 1999.
- [11] <http://tstat.tlc.polito.it/tsol/riassunto.php>